ADAPTable Sensor Systems Phase 2
STRATEGIC TECHNOLOGY OFFICE DARPA-BAA-12-61

Topic 2: Reusable Core Software

Distributed Synchronization Software for the Sensor Nodes
Final Report

Michigan Technological University
PI: Saeid Nooshabadi
(saeid@mtu.edu)

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
MAR 2015		**00-00-2013 to 00-00-2015**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Reusable Core Software Distributed Synchronization Software for the Sensor Nodes	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Michigan Technological University,,Departments of Electrical & Computer Engineering,,and xComputer Science,,Houghton,,MI**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**12**	

Table of Contents

MILESTONE ACCOMPLISHMENTS

This project has achieved several milestones consistent with the Statement of Work.

Milestone 1 (May 2013 to July 2013): We developed a synchronization algorithm that estimates the phase offset of one pulse per second (1PPS) clock derived from the TXCO with respect with a noisy one pulse per second (1PPS) GPS reference signal. The accuracy of this algorithm is about 0.5 ms over 3 minutes of data collection. The code has been ported to the ADAPT platform. This algorithm was 4 times better the previous implementation.

Milestone 2 (August 2013 to December 2014): We developed a self-calibration algorithm that accurately estimates the change in the TXCO oscillator frequency from the nominal 19.2 MHz. This algorithm uses the noisy 1 PPS GPS data measurements over a period of over 15 hours to estimate the drift in TXCO oscillator frequency. The estimated accurary of this frequency estimation algorithm is approximately 1 Hz. The implementation code has been uploaded to the ADAPT repository.

Use of this algorithm eliminates the need for expensive in-factory calibration. This self-calibration step needs to be applied just once before the in-field operation of the node. There might be a need to apply the self-calibration step if there is a significant drift (beyond) one clock cycle due to aging.

Milestone 3 (January 2014 to February 2014): We developed a fractional frequency adjustment algorithm that adjusts for fractional drift in the TXCO oscillator frequency. This frequency adjustment is continuously applied to maintain an accurate 1PPS clock for the ADAPT board. This accuracy of this algorithm is within 0.5ms over 24 hours of operation. In most node-to-node observations the clock inter-node clock offset is bounded and is no more than 0.5 ms over 8 hours. With this algorithm, the major source of inter-node clock offsets is no more the TXCO clock offset and instead shifts to transitory variations in the transmitter/receiver system. The C code for this algorithm will be ported to the ADAPT board soon. This algorithm is 16 times better than the previous implementation.

Milestone 4 (February 2014 to present): We are developing a frequency maintenance algorithm that continuously recalibrates the TXCO without the need for re-application or self-calibration due to aging. This algorithm will use the noisy 1 PPS GPS data measurements over a long period of time to build a feedback loop, similar to what one might find in a phase-lock loop that keeps the TXCO clock oscillator phase within 0.5 ms of the true time phase.

We make adjustments to the phase and frequency with each GPS observation using an iterative Kalman filter process. The standard deviations of the phase and frequency of the TCXO system should be upper bounded by standard deviation of GPS measurements (±0.5 ms) and TCXO frequency shift (one ppm), respectively.

To perform Kalman filtering we first develop a discrete time model for clocking system. In our model, we pretend that the TCXO is a perfect representation of ”true time” and use it to track the phase and frequency of the GPS clock. Even though the truth is exactly the opposite, the estimated phase and frequency of the GPS clock relative to the RTC can still be used to update the phase and frequency of the RTC, thus keeping the two aligned.

Deliverables:

Code as uploaded to the ADAPT Software Repository:

writeCal.c -- source code for a program that can write a calibration file on a non-node device for the purpose of uploading to an ADAPT node.

readCal.c -- source code for a program that can read a calibration file downloaded from an ADAPT node on a non-node device.

processRtc.sh -- a shell script that will process the rtc logs and generate some data files

rtc.c -- the main component of the rtc program, handles any changes or adjustments to the TCXO. Modification to this file includes adding capability for fractional frequency, loading and storing the current calibration from/to a file, and new technique to find the GPS time marker for the purposes of minimizing the impact of GPS noise on the system.

rtcmaint.c -- a secondary source file for the rtc program. Contains all the code for initial self-calibration, and any corrective algorithms that have been created. These corrective algorithms change the TCXO phase and/or frequency on the ADAPT node. Currently there are three 1) GPS direct phase skew 2) linear regression phase and frequency adjustment 3) kalman filter based.

Appendix A: MILCOM Paper
Node Synchronization in a Wireless Sensor Network Using Unreliable GPS Signals

Daniel R. Fuhrmann*, Joshua Stomberg§, Saeid Nooshabadi*§
Dustin McIntire†, William Merill†
*Departments of Electrical & Computer Engineering, and §Computer Science
Michigan Technological University
Houghton, Michigan, Email: {fuhrmann,jcstombe,saeid}@mtu.edu
†Silver Bullet Technology, {dustin,william.merill}@silver-bullet-tech.com

***Abstract*— This paper presents our findings in using pulse measurements from a jittery one pulse per second (pps) global positioning system (GPS) clock, to synchronize the real-time clock (RTC) in each node of a wireless sensor network, when the timing jitter is subject to a empirically determined bimodal non-Gaussian distribution. Specifically, we 1) estimate the RTC phase and align it with an estimate of the true time phase, 2) calibrate the frequency of a 19.2 MHz low-cost temperature-compensated crystal oscillator (TCXO) that drives the one pps RTC, and 3) track and compensate TCXO frequency variations due to environmental and aging effects. In our GPS-driven synchronization methodology we adopt a statistical signal processing framework to estimate the 2% percentile in the bimodal distribution, perform a long-term frequency calibration with fractional frequency adjustment, and track the changes in the TCXO frequency to within three tick per second over a nominal 19.2 MHz frequency with an adjustment made every four hours.**

***Index Terms*— clock synchronization, GPS, wireless sensor networks, Kalman filtering**

I. Introduction

Wireless sensor networks (WSNs) for intelligence, surveillance, and reconnaissance (ISR) infrastructures systems are considered fundamental for homeland security, detection of chemical and biological threats, and other similar application [1] (and references therein).

In 2012 the Defense Advanced Research Projects Agency (DARPA) embarked on the ADAPTable Sensor System (ADAPT) program that seeks to investigate innovative approaches to building ISR Systems. ADAPT uses commercial practices to accelerate the development and delivery of ISR systems without burdening the resulting systems with the limitations typically found in the military use of commercial technology (*e.g.*, limitations in performance, security, reliability or assured operation.)

This research was supported by the DARPA ADAPTable Sensor Systems Phase 2 Contract No. HR0011-13-C-0057. The views, opinions, and/or findings contained in this article are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense. Approved for Public Release, Distribution Unlimited.

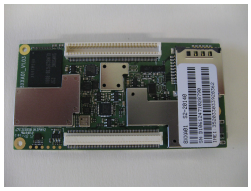

Fig. 1. ADAPT Board

Fig. 2. ADAPT Blue Node

Towards that aim ADAPT prototype nodes (Figs 1 and 2) have been developed using of-the-shelf components. The communication scheduling protocol among the ADAPT nodes, like many WSNs, is based on time division multiple access (TDMA), where the communication time slots are buffered with guard times. To efficiently use the transmission bandwidth the guard time needs to be kept to a minimum. This in turn requires tight inter-node clock synchronization.

Clock synchronization provides a common notion of time across the nodes in the WSN formation. It is crucial for the correct functioning of the WSN and is required for fundamental operations such as data fusion, power management, transmission scheduling [1], [2], [3], [4], [5]. For a computer system the relation between the approximation of real-time clock (RTC) $T(t)$ in relation to the timer hardware clocked by the TCXO can be written as

$$T(t) = ct + p \tag{1}$$

where c and p correspond to the drift rate and phase offset, respectively, of the RTC. For a perfect RTC, the drift $c = 1$. The quantity c directly relates to the difference between the TCXO's calibrated/nominal frequency and its actual value. This value can be quite large, up to a few parts per million

(ppm), which can in turn lead to a drift in the RTC phase of a few hundreds of ms over the course of an hour.

In a WSN consisting of several nodes, a synchronization algorithm aims to preserve a global timescale throughout the network at all times. Maintaining synchronization across the nodes requires correcting the clock rates c as well phase offsets p. If c is not corrected for, the RTCs drift and the offset needs to be corrected repeatedly to keep the RTCs synchronized over a time period.

In most WSNs, RTC synchronization simplifies to maintaining relative clocks among the nodes [6], [7], [8]. In this scheme, each node maintains its own independent local clock. However, each keeps a table of its relative drift and offset of its clock with respect to clocks on the other nodes. That way at any given time, the local time of the node can be converted to some other node's local time and vice versa.

One approach to RTC synchronization is through the frequent exchange of timing information among the nodes. However, inter-node communication imposes high demand on the power budget. According to [9] for a typical wireless node in a WSN, the energy to transmit one kilobit (kb) of data more than 100 meters is about the energy required to execute three millions of instructions on an embedded processor. Reducing the frequency of the synchronization among the nodes demands a tighter control on rate drift c on individual nodes. This in turn requires a good tracking of TCXO frequency drift due to environmental variations such a temperature and aging, and provision for appropriate compensation to mitigate or at least minimize its effect.

The Qualcomm MSM processor on the ADAPT board in addition to containing a dual core CPU, is equipped with a GPS DSP primarily for the purpose of location detection. GPS has been widely used in base stations for extraction one-pulse per second (pps) timing information, synchronized to GPS or UTC within 15 ns (one sigma). However, GPS DSP produces software conditioned timing pulses approximately once per second which synchronizes to GPS with a sigma as large as two milliseconds (ms). In this work we use GPS DSP hardware in the MSM processor to maintain a tight control on the drift c, and to phase compensate the individual nodes to align them to the same global GPS clock. With this technique all nodes in the WSN synchronize to the same global clock to within ± 2 ms. Further, relative synchronization for TDMA protocol can also be extended to nodes without GPS signals through inter-node exchange of timing information.

The organization of the paper is as follows. Section II presents the characteristics of the GPS one pps clock. Section III presents the strategy to extract reliable timing information from the unreliable GPS signal. Section IV discusses the strategy to use the GPS one pps signal to calibrate the TCXO on each node prior to deployment in the field. Section V presents a Kalman filter technique to track the gradual drifts in the TCXO frequency due the environmental variations in the field, and to mitigate its effect. Section VI provide a discussion on the limitation of our strategy in extracting the GPS timing information.

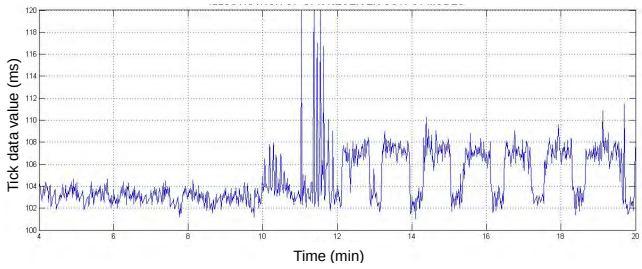

Fig. 3. Illustration of GPS receiver output modes on ADAPT node

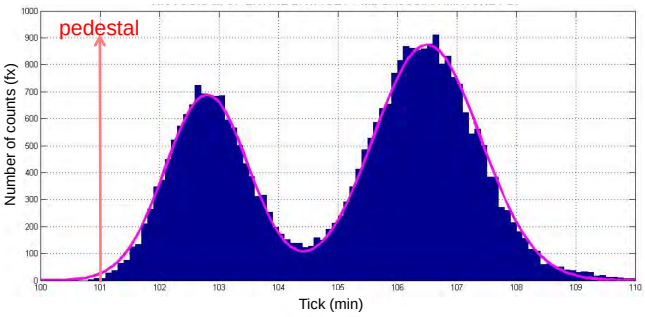

Fig. 4. Histogram of probability distribution function of GPS one pps signals

II. GPS PULSING CHARACTERISTICS

Fig. 3 illustrates the timing of the one pps GPS pulse with respect to beginning of a one-second RTC period, over 20 minutes of observation. Nominally one would like to see the GPS pulses align with the 100 ms point within the RTC period. As seen the behavior can be characterized as a period of noisy early arrival (102 to 104 ms), a burst of late arrivals ($>$ 110 ms), followed by a periodic alteration between early and late mid-range arrivals (104 to 108 ms) with a periodicity of about 90 seconds. In other nodes we found sporadic instances where the period would exceed 300 seconds. Fig. 4 depicts the empirical histogram of the GPS pulse arrival times for these 1200 samples.

III. TIMING EXTRACTION FROM GPS

The bimodal distribution in Fig. 4, while interesting, does not help to extract useful timing information through aver-

aging, since we have not observed a consistent probability density function (pdf) over multiple nodes and multiple datasets. However, what we have observed consistently in all experimental datasets is that the support of the empirical pdf has an approximate lower bound, that is to say, the empirical pdf is heavy-tailed on the right but not on the left. A reasonable approach in this case is to model the timing jitter as a random variable whose density has a support region with a strict lower bound, plus additive Gaussian noise with small variance. This suggests that a small percentile (e.g. 2%) point in the histogram can provide a robust estimate of the GPS "true time" that is insensitive to the right-tail distribution. We refer to this 2% percentile in the histogram as the "pedestal" or time marker in the pulse signal since 98% of the data values lie above it. Algorithm 1 presents the process of identification of the pedestal point.

Algorithm 1 Identification of the pedestal point in Figure 4

```
1:
2:  procedure PEDESTAL ESTIMATION
3:      [Sliding Window] Create a sliding
4:          window of certain size.
5:      [Choose Step Size] Choose a step size for
6:          stepping through the over-lapping windows.
7:      [Histogram] For each step create
8:          histogram of the data on one millisecond
9:          bins.
10:     [Bin Identification] Identify the 20 consecutive
11:         bins that contain the largest number
12:         of counts (account for wraparound if
13:         necessary.)
14:     [Second Percentile] Use second percentile
15:         data point within the 20 milliseconds
16:         as the estimator of the pedestal level,
17:         or time marker, identified in analysis of
18:         empirical data.
19:
20: end procedure
```

Algorithm 1 was tested on various window sizes across entire one-hour datasets. We used sliding window sizes of 30 seconds to 10 minutes, with a step size of 30 seconds. Subsequently the mean and standard deviation of the pedestal values across the full captured empirical datasets were computed. Application of algorithm across the full datasets yielded the results shown in Fig. 5.

We observe that the ADAPT GPS receivers produce one pps pulses with the the pedestal point settles at a mean of 102 ms, and standard deviation of 0.5 ms in about two to three

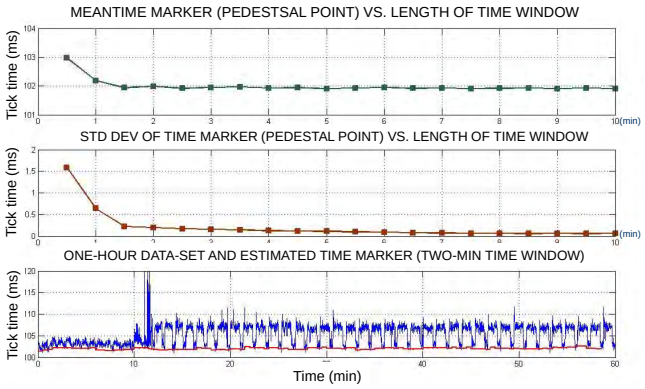

Fig. 5. Synchronization of ADPAT node with GPS signals, (a), The mean value (in ms) with respect to the beginning of a second, (b) standard deviation (in ms), (c) data-set values and actual 2-minute time marker or time marker (in red)

minutes. It is, therefore, possible to synchronize the RTC with the GPS clock, to within in less than three minutes. The estimated pedestal value is used to adjust the phase of the RTC. Note that (thus far) we make no attempt to correct for the frequency drift, if any exists. Here it is assumed that the one pps RTC clock and the 1pps GPS clock are operating at the same frequency and that the pedestal values are used only to align the phases.

IV. GPS ASSISTED SELF-CALIBRATION

A. Use of GPS for self-calibration

To keep the frequency drift to within a fraction of the specification of one ppm of variation, a typical low-cost TCXO requires post-manufacturing and regular in-field calibration. However, manual calibration in the laboratory is cumbersome and expensive. In this section we utilize the algorithm from the last section to automate the TCXO calibration. Fig. 6 depicts the manifestation of RTC drift in the GPS observations for an uncalibrated node. The RTC drift in this figure is due to difference in the real operating frequency of the TCXO and the nominal frequency used in programming the RTC timer registers. In Fig. 6 one long data record ($\sim$ 10 hours) is broken into five-minute segments (300 seconds). The phase estimation (pedestal finding) technique from Algorithm 1 from the previous section is applied to each segment. Note that in this self-calibration mode the computed phases are not used to adjust the phase of RTC, but rather they are allowed to drift. They can then be fit to a straight line, the slope of which determines the frequency offset from nominal. The line-fitting method we propose finds the straight line that minimizes the sum of the absolute values of the distance from the points to the line, subject to the constraint that the

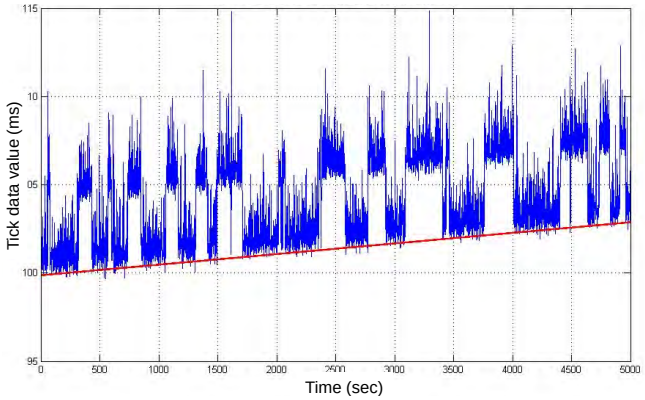

Fig. 6. The drift in the GSP observation in an uncalibrated node

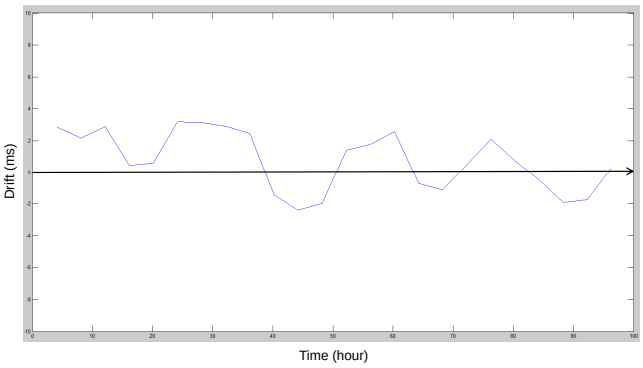

Fig. 7. GPS phase estimation appears as noise with perfect frequency calibration

Fig. 8. Fractional frequency adjustment for RTC

line lie below all the date points. This is a straightforward convex optimization problem, the solution to which requires the calculation of the lower convex hull of the 100 or so data points. The two points on the convex hull nearest the time midpoint define a line, and the slope of that line becomes the estimate of the frequency offset. With the frequency offset in hand the ADAPT board makes an adjustment to the timer register to set the drift in (1) to a value close to unity.

B. Fractional frequency adjustment

Since the timer counters can only accept integer values, self-calibration leaves a fractional residual frequency error. We make adjustments to timer register value for RTC by alternating between two frequencies proportional to the fractional residual error. Fig. 8 depicts the technique for a residual error of 0.665 TCXO tick. With this the drift in RTC should be very close to zero, and the requirement for synchronization with GPS becomes less frequent. With nearly perfect calibration GPS-based phase error appears as noise with a mean value of zero as depicted in Fig. 7.

It should be noted that clock synchronization with fractional frequency correction is nearly perfect when the node is properly calibrated (self-calibration or factory calibration). However, there is no guarantee of perfect calibration. In our experimentation, even with our best calibration effort there can be a residual inter-node frequency difference of up to one TCXO tick per second. This corresponds to an phase offset of about 0.2 ms in one hour. Further, due to environmental variations, primarily temperature, (both seasonal and daily variations), and aging, even with best calibration the RTC diverges with time, albeit very slowly. We have noticed a worst inter-node drift of 0.5 ms per hour. Therefore, GPS-based phase alignment (or over the air-alignment) is required at regular intervals to adjust the RTC phase and keep the nodes synchronized. Periodic recalibration at intervals on the order of one or more days is required to capture variation in the TCXO frequency due to environmental and aging effects.

V. Kalman filtering for drift calibration

As was mentioned in the previous section, occasional recalibration is required to capture variation in the TCXO frequency due to environmental and aging effects. Alternatively, we can make adjustments to the phase and frequency with each GPS observation using an iterative Kalman filter process. The standard deviations of the phase and frequency of the TCXO system should be upper bounded by standard deviation of GPS measurements (± 0.5 ms) and TCXO frequency shift (one ppm), respectively.

To perform Kalman filtering we first develop a discrete-time model for clocking system. In our model, we pretend that the TCXO is a perfect representation of "true time" and use it to track the phase and frequency of the GPS clock. Even though the truth is exactly the opposite, the estimated phase and frequency of the GPS clock relative to the RTC can still be used to update the phase and frequency of the RTC, thus keeping the two aligned.

The state vector for the dynamic system is a 2×1 vector comprising the phase and frequency of the GPS clock. The dynamic model of the system can be formulated as

$$\mathbf{S}_{\text{TCXO}} = \begin{bmatrix} p \\ f \end{bmatrix} = \begin{bmatrix} Phase\ Difference \\ Frequency\ Drift \end{bmatrix} \qquad (2)$$

where

$$\mathbf{S}_{\text{TCXO}}(k) = \mathbf{A}\mathbf{S}_{\text{TCXO}}(k-1) + \mathbf{B}u(k)$$

with

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad u(k) = \texttt{i.i.d.}\ \mathcal{N}(0, \sigma_{df}^2),$$

where σ_{df} is the standard deviation on the rate of change of TCXO frequency in Hz/second.

We do not use individual GPS pulses as observations for the Kalman filter but rather the pedestal points computed for datasets of several hundred pulses, observed at intervals on the order of 2-4 hours. The pedestal points are modeled as Gaussian random variables. The model for the GPS observation can be described as,

$$\mathbf{S}_{\text{GPS}}(k) = \mathbf{C}\mathbf{S}_{\text{TCXO}}(k-1) + \mathbf{D}v(k) \tag{3}$$

with

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1 \end{bmatrix}, \quad v(k) = \texttt{i.i.d.}\ \mathcal{N}(0, \sigma_p^2),$$

where σ_p is the standard deviation in the variation in the phase of the GPS observations in seconds.

The initial state distribution is assumed to be Gaussian with mean $\boldsymbol{\mu}$ and covariance $\mathbf{R}$:

$$\mathbf{S}_{\text{TCXO}}(0) \sim \mathcal{N}(\boldsymbol{\mu}(0), \mathbf{R}(0)), \tag{4}$$

with

$$\mathbf{R}(0) = \begin{bmatrix} r_{00}(0) & r_{01}(0) \\ r_{10}(0) & r_{11}(0) \end{bmatrix} = \begin{bmatrix} \sigma_{p0}^2 & 0 \\ 0 & \sigma_{f0}^2 \end{bmatrix}, \quad \boldsymbol{\mu}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

where σ_{p0} and σ_{f0} correspond, respectively, to the standard deviation in the initial phase in seconds, and frequency in Hz of TCXO.

The steady-state behavior of the Kalman filter is not sensitive to the initial state distribution, which can be chosen somewhat arbitrarily.

The Kalman filter process updates the mean and covariance of the state distribution $\boldsymbol{\mu}(k)$ and $\mathbf{R}(k)$ based on the dynamic system model and the observation model:

$$\mathbf{S}_{\text{TCXO}}(k) \sim \mathcal{N}(\boldsymbol{\mu}(k), \mathbf{R}(k)) \tag{5}$$

with

$$\mathbf{R}(k) = \begin{bmatrix} r_{00}(k) & r_{01}(k) \\ r_{10}(k) & r_{11}(k) \end{bmatrix},$$

$$\boldsymbol{\mu}(k) = \mathbf{A}\boldsymbol{\mu}(k-1),$$

and

$$\mathbf{R}(k) = \mathbf{A}\mathbf{R}(k-1)\mathbf{A}^{\text{T}} + \mathbf{B}\sigma_{df}^2\mathbf{B}^{\text{T}}$$

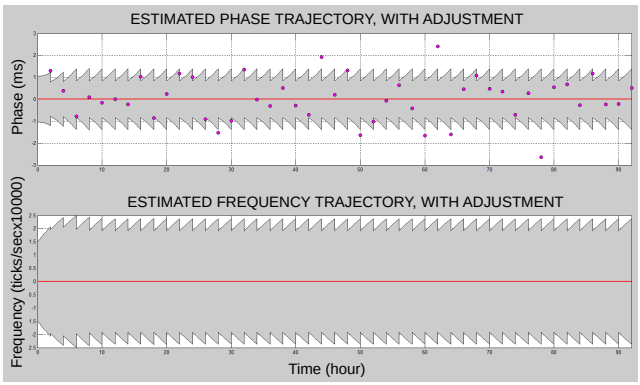

Fig. 9. Kalman filtering to adjust the RTC phase and TCXO frequency. The dark bands indicate ± one standard deviation in estimation of phase and frequency drift.

The updates to the state distribution parameters after each GPS observation are given by

$$\boldsymbol{\mu}(k) \leftarrow \boldsymbol{\mu}(k) + \mathbf{R}_{\text{XY}}\mathbf{R}_{\text{YY}}^{-1}(\mathbf{S}_{\text{GPS}}(k) - \mathbf{C}\boldsymbol{\mu}(k)) \tag{6}$$

and

$$\mathbf{R}(k) \leftarrow \mathbf{R}(k) - \mathbf{R}_{\text{XY}}\mathbf{R}_{\text{YY}}^{-1}\mathbf{R}_{\text{YX}}$$

with

$$\mathbf{R}_{\text{YY}} = \mathbf{C}\mathbf{R}(k)\mathbf{C}^{\text{T}} + \mathbf{D}\sigma_p\mathbf{D}^{\text{T}},$$

$$\mathbf{R}_{\text{XY}} = \mathbf{R}(k)\mathbf{C}^{\text{T}}$$

and

$$\mathbf{R}_{\text{YX}} = \mathbf{C}\mathbf{R}(k)$$

The processing of on real data in Fig. 9 shows how the Kalman filter makes regular adjustment to RTC phase and TCXO frequency to maintain an upper limit to phase shift. The bands in both figures indicate ± one standard deviation in the estimation of phase and frequency, as determined by the covariance of the state distribution. Fig. 10 depicts actual observation of Kalman filter making correction to TCXO phase and frequency to within three tick per second, with one GPS observation and adjustment made every four hours, with a maximum drift of 2.5 ms within that many hours.

VI. Discussion on pedestal point

In Section III we modeled the pedestal point as the 2% percentile point of a distribution with zero support below 102 ms and uncertain right tail, plus additive Gaussian noise. The distribution of the resulting pedestal point was modeled as Gaussian random variable with mean 102 ms and standard

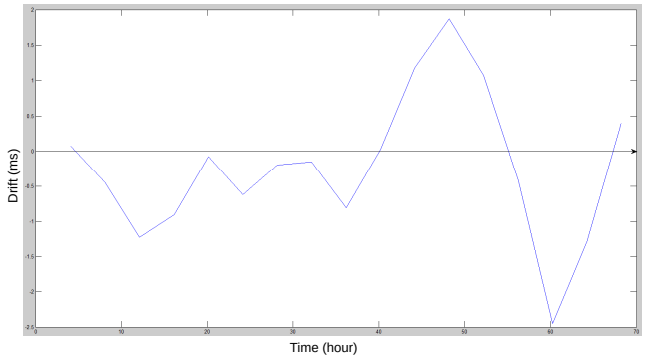

Fig. 10. Observation of Kalman filtering on the ADAPT board making adjustment to the RTC phase and TCXO frequency.

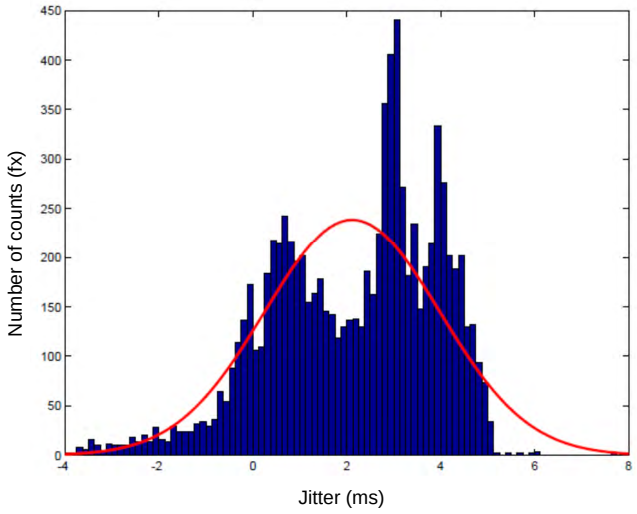

Fig. 11. Distribution of pedestal point and its equivalent Gaussian distribution model of the same pdf. The jitter is measured with respect to the 2 percentile point of 102 ms indicated in Fig. 5.

deviation 0.5 ms (see Fig. 5), and this model was used in the design of the Kalman tracking filter. In reality, not all the ADAPT nodes produce timing data that is consistent with this model. Fig. 11 shows the histogram of about 9000 estimates of the pedestal point, obtained through Algorithm 1, for one particular ADAPT node whose behavior deviates from the model described above. Superimposed on the histogram is the pdf for a Gaussian random variable with the same mean and variance. As can be observed, the mean exhibits a bias of about 2 ms (i.e. the mean pedestal point is approximately 104 ms) and the standard deviation is about 2 ms. When data drawn from an ADAPT node with this behavior is used to drive the Kalman tracking filter, the drift in the TCXO phase increases to more than 4 ms per every 4 hours. However,

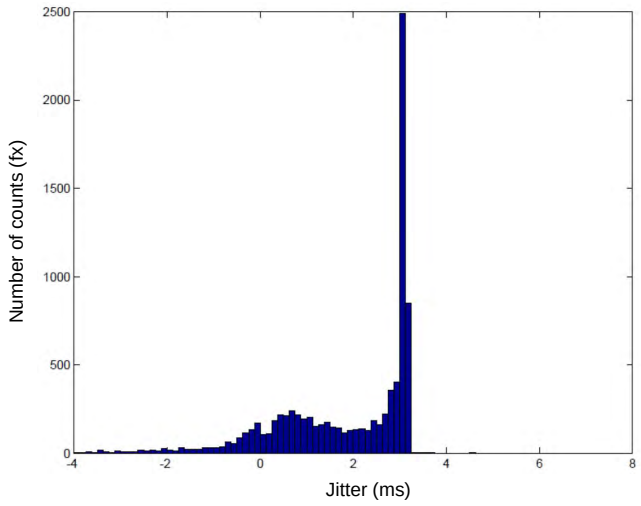

Fig. 12. Distribution of pedestal when skewed. The jitter is measured with respect to the 2 percentile point of 102 ms indicated in Fig. 5.

it is possible to make the Kalman tracker more robust with respect to these kinds of uncertainties in the data distribution. We have introduced a heuristic pointwise nonlinearity that is applied to the GPS PPS data prior to using it to update the Kalman filter. Our heuristic nonlinearity is a soft limiter that saturates at an error of 3 ms above the current estimate of the oscillator phase. A histogram of the output of this limiter, driven by GPS data with distribution shown in Fig. 11, is shown in Fig. 12. Using this soft limiter is a simplified alternative to the full Bayesian nonlinear filter that would take into account the bimodal distribution for the GPS PPS shown in Fig. 11. This soft limiter limits the the drift in the TCXO phase to no more than 2.5 ms per every 4 hours.

VII. Conclusions

In this paper we applied statistical signal processing techniques on a jittery one pulse per second (pps) global positioning system (GPS) clock on a Qualcomm MSM processor, to make accurate phase adjustments to the RTC on the nodes in a WSN. We also employed the GPS signals to estimate the frequency drift on the nodes and perform accurate (to within 0.05 ppm) TCXO frequency self-calibration on the nodes. Further, we used a Kalman filtering technique to track the changes, and make correction, to TCXO phase and frequency to within three tick per second ((0.5 ms per hour) over a nominal frequency of 19.2 MHz.

References

[1] Y.-C. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, Jan 2011.

[2] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July 2004.
[3] R. Carrano, D. Passos, L. Magalhaes, and C. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 181–194, First Qrt 2014.
[4] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 3, pp. 281–323, 2005.
[5] S. Lasassmeh and J. Conrad, "Time synchronization in wireless sensor networks: A survey," in *Proceedings of the IEEE SoutheastCon (SoutheastCon)*, Charlotte, NC, March 2010, pp. 242–245.
[6] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, Dec. 2002.
[7] J. van Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks," in *Proceedings of ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, San Diego, CA, 2003, pp. 11–19.
[8] M. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," in *Proceedings of IEEE Wireless Communications and Networking, (WCNC 2003)*, vol. 2, New Orleans, LA, March 2003, pp. 1266–1273.
[9] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Comm of ACM*, vol. 43, no. 5, pp. 51–58, May 2000.